**Arabic Variant TLD Issues and Requirements**

1. **Background**

This document has been prepared to identify issues about how variants may be defined, managed and implemented in new IDN gTLDs. .

In 2009, an independent implementation working team was formed after discussions during the ICANN meetings in Mexico City and Sydney to study these issues. The team included linguistic and technical experts from various language communities, and was co‐chaired by two ICANN Board Directors who are well‐versed in the fields of IDN and DNS. The team recommended that variants not be delegated as TLDs at that time, and that if desired variants are to be delegated, certain conditions must be fulfilled[[1]](#footnote-1).

To develop potential solutions for the delegation of IDN variant TLDs, the ICANN Board in its 2010 meeting[[2]](#footnote-2) in Norway directed the CEO to:

… develop (in consultation with the ICANN Board ES-WG) an issues report identifying what needs to be done with the evaluation, possible delegation, allocation and operation of IDN gTLDs containing variant characters, as part of the new gTLD process in order to facilitate the development of workable approaches to the deployment of gTLDs containing variant characters IDNs. The analysis of needed work should identify the appropriate venues (e.g., ICANN, IETF, language community, etc.) for pursuing the necessary work. The report should be published for public review.

Accordingly, ICANN in consultation with community has proposed to conduct as many as six case studies in the following scripts (Arabic, Chinese (Traditional and Simplified), Cyrillic, Devanagari, Greek, and Latin) to investigate the set of issues that need to be resolved to facilitate a good user experience for IDN variant TLDs. Each case study team has been requested to submit an Issues report for the script it is looking into. From these Issues report, a single Issues Report will be created.

This document presents the Issues report developed by the Arabic Case Study Team. It is worth to mention that even these issues maybe of some concerns to the registries, this document is solely intended for the TLD level.

1. **Introduction to Arabic Script**

Arabic script has been used across North Africa, Middle East, Central Asia South Asia and South East Asia to write multiple languages from Semitic, Indo-Iranian, Indo-European, Dravidian, Turkic, African and and Austronesian language families. Some salient areas Arabic script is currently being used are highlighted in the illustration below.

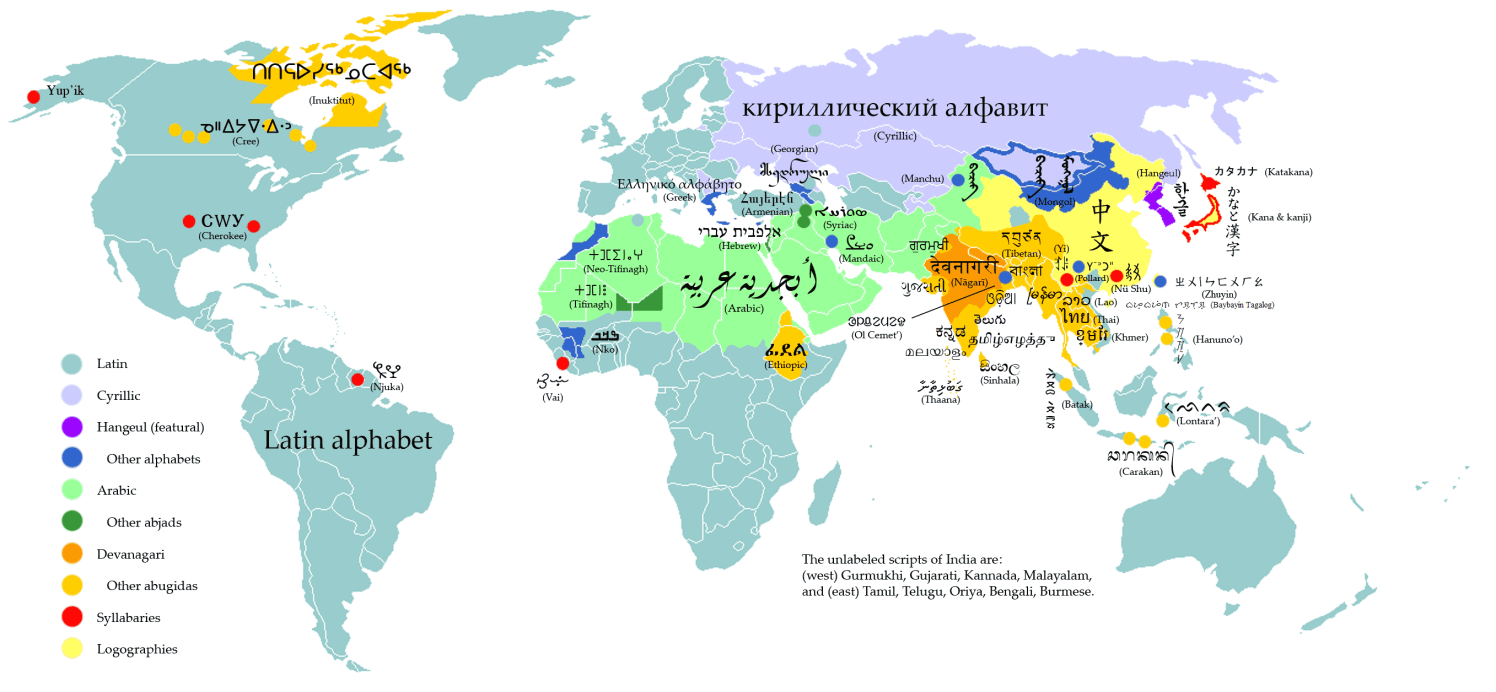


Figure 1: Writing Systems of the World

(Source: <http://en.wikipedia.org/wiki/File:WritingSystemsoftheWorld4.png>)

Arabic writing system is also referred to as Abjad system, in which consonantal sounds are represented as base characters and vowels are normally represented by optional combining marks on these base characters. The writing system is cursive and each letter may have multiple shapes, generally categorized as initial, medial, final and isolated forms, based on where it occurs within the connected portion of a (sub)word, called a ligature, or whether it occurs by itself (not joined with any other letter).

An additional complexity in the Arabic script is the bi-directionality of the script. The script is generally written from right to left, however, the digits are written from left to right. Though this does not change the key-press order of input into computing devices, it may have significant impact on how the input is displayed. Further complexity is introduced if the (right to left) Arabic script is mixed with other (left to right) scripts, like Latin.

|  |  |
| --- | --- |
| Arabic script has a variety of writing styles, with Naskh predominantly being used to write Arabic language and Nastalique and Farsi being used for some of the other languages using Arabic script. Additional writing styles, which are used for stylistic reasons, include Thuluth, Diwani, Kufi, Riqa, etc. Some of these writing styles are illustrated in Figure 2 (with same Arabic language phrase written in different styles). The various styles show both the base characters and the optional combining marks. Thuluth and Diwani also include non-linguistic ornate marks. |  |
| Figure 2: Arabic Script Writing Styles: Naskh, Nastalique, Kufi, Thuluth, Diwani and Riqa  (Source: Hussain 2004) |

1. **General Principles about the Report**

While discussion, the Arabic Script Case Study Team deliberated on many issues and has generally agreed on the following three general principles for interpreting the eventual document.

As per the scope of the study, the Arabic Script Case Study Team agreed to talk generally for TLD space, without making the distinction between ccTLD or gTLD (and specify where our recommendations or comments may diverge).

The team also agreed to limit the scope of work to TLDs (not second or other level labels), unless the recommendations apply to all levels (where it has been made explicit).

Finally, though the team is generally confident on the identification of the issues and any corresponding recommendations, some issues may still need to be discussed with representatives of languages communities not represented in the committee (e.g. use of Arabic script in African languages). In such cases, it has been explicitly identified for further consultation.

1. **Terminology**
2. **TLD Label Valid Code Points for Arabic Script**

The Arabic script characters are encoded from U+0600 – U+06FF and U+0750 – U+077F[[3]](#footnote-4) in the Unicode standard. Only a subset of these characters are PVALID, i.e. allowed for use in labels (e.g. see RFC 5892). The team suggests further limiting the use of these characters for TLDs, as per the following details.

1. 200C (ZWNJ): an issue as they are CONTEX-J but should not be allowed for TLDs
2. 200D (ZWJ): Not needed in Arabic script
3. 0610-061A: an issue as they are PVALID but should not be allowed for TLDs
4. 0621-063F: OK, PVALID and needed for TLDs
5. 0641-064A: OK, PVALID and needed for TLDs
6. 064B-065F: an issue as they are PVALID but may not be allowed for TLDs
7. 0660-0669: an issue as they are PVALID but should not be allowed for TLDs because digits
8. 066E-0670: an issue as they are PVALID but should not be allowed for TLDs
9. 0679-06D3: OK, PVALID and needed for TLDs
10. 06D5: OK, PVALID and needed for TLDs
11. 06D6-06DC: an issue as they are PVALID but should not be allowed for TLDs
12. 06DF-06E8: an issue as they are PVALID but should not be allowed for TLDs
13. 06EA-06ED: an issue as they are PVALID but should not be allowed for TLDs
14. 06EE-06EF: OK, PVALID and needed for TLDs
15. 06F0-06F9: an issue as they are PVALID but should not be allowed for TLDs because digits
16. 06FA-06FF: OK, PVALID and needed for TLDs
17. 0750-077F: OK, PVALID and needed for TLDs
18. FE73: an issue as they are PVALID but should not be allowed in any label (TLDs and other labels)

This list is may change as Unicode standard changes and more characters are added to Arabic script, as per the IDNA 2008 protocol specifications.

Notes Regarding Space Marks:

As per the above recommendations, combining characters are recommended not to be allowed for TLD labels. This is due to the fact that there are different kinds of optional marks in Arabic script, and their usage differs across languages. As they are optional and there can be multiple optional marks per letter in a label, their use may create a very high number of variants. Additionally, due to their small size (as glyphs) and optionality, they can also cause a high degree of user confusion and security issues. Their use should therefore be very carefully regulated. Hence, it is recommended that they are not allowed for the TLDs. Please note that the identical character variants may also be caused by a letter combining with a combining mark. Such cases are given in Appendix A.2 (A.2.1 and A.2.2). Some of these cases are addressed automatically through the IDNA 2008 protocol specifications, which requires the characters to be composed, if not presented in such form. However, such composition only takes place, where equivalence has been defined by the Unicode standard (Davis and Durst 1999). In cases where composition has not been defined by Unicode (for various reasons), such cases would need to be explicitly managed as variants (status listed in the final column of Appendix A.2.

As discussed, this also depends on feedback from communities using the Arabic script for African languages, as encoded by the Unicode standard. Appendix A.2 may still be useful reference for non-TLD label formation.

1. **Character Variants in Arabic Script**

Variant labels in Arabic script may occur due to reasons motivated by linguistics, stylistics of writing styles or encoding. The current issues document lists all possible cases where variants may occur, for consideration and further stipulation of how variant sets may be constituted. The current analysis distributes character variants into four categories: identical, confusingly similar, interchangeable and optional.

1. **Identical Cases**

In these groups of letters, a letter exhibits an identical shape in at least one of their initial, medial, final or isolated forms with at least another letter in the group. The groups are formed transitively, i.e. if A is a character variant of B, and B is a character variant of C, then A is also considered a character variant of C, even if the condition of being identical is not met in A and C (e.g. see the Hay set). These groups are defined in Appendix A.1.

* 1. Kaf group – limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request)
  2. Heh group – limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request)
  3. Yeh group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request)
  4. Feh group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request)
  5. Veh group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request)
  6. Teh marbuta group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request)
  7. Heh hamza group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request)
  8. Ttey/Rnoon group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request)
  9. Noon group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request)
  10. Theh group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request) (confusable with pay, not Thay)

1. **Similar Cases**

In these groups of letters, a letter is confusably similar in shape in at least one of their initial, medial, final or isolated forms with at least another letter in the group. The groups are formed transitively. These groups are defined in Appendix B.

* 1. Kaf group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request). This is needed as the Swash Kaf version is interchangeably used with other Kafs and the former is considered the same letter, with just stylistical variation. It is considered different from other Kafs only in Sindhi language, for which it has been encoded.
  2. Yeh group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request). The addition Yeh in this case has very slight tail at its end which may be considered a stylistic variation and be confused with regular Yeh by most users of Arabic script, except speakers of Pashto who distinguish it from other Yehs.
  3. Alif Hamza above group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request). The wavy Hamza is not distinguishable by most Arabic script users.
  4. Alif Hamza below group - limit as one at TLD level; all are possible for TLD registration (no preferred over other, depends on registrant request). The wavy Hamza is not distinguishable by most Arabic script users.

Dot orientation could cause confusion among Arabic script users and thus can be a potential candidate for character variants. Such confusing cases have been listed in Appendix B.2. It should be investigated further with feedback from relevant language communities (not represented on the committee) for resolution.

Additionally there are some more cases, which are generally not confusable by Arabic script users, but may become confusable in small font size, especially in fonts which are not very accurate in shaping. These include the cases in Appendix B.3.

1. **Interchangeable Cases**

In some cases, either similar sound, common different presentations or pronunciations, or usability across different languages sometimes cause different characters to be used interchangeable. This will cause a user confusion and should be treated as variants. .

* 1. Combining Hamza – All forms of ALEF (ALEF, ALEF WITH MADDA ABOVE, ALEF WITH HAMZA ABOVE, and ALEF WITH HAMZA BELOW) are commonly used in mixed. For examples, ALEF is used instead of all of the other forms (e.g., ALEF WITH HAMZA ABOVE or ALEF WITH HAMZA BELOW), ALEF WITH HAMZA ABOVE is used instead of ALEF WITH MADDA ABOVE. Part of the writing “confusion” depends sometimes on how it is pronounced.
  2. The Teh Marbuta (U+0629) and Heh (U+0647) are used interchangeably at the end of a word by speakers of these languages, because they sound same in the context. Thus, these may cause user confusion in the labels.

1. **Label Generation Policy**

ICANN would adopt a single “Arabic Script TLD table (ASTT)”, which it would use to determine (by ICANN) if a TLD label and its variants are valid.

The team agrees that the structure proposed in JET Guidelines (RFC 3743) is not appropriate for Arabic script. There are many differences, e.g. only a two column format may be sufficient and the relationship between variants is symmetric in Arabic script based labels. Thus a separate effort involving the Arabic Script community needs to be undertaken to consider the structure of Label Generation Policy for Arabic script. Defining the structure is important for effective use, reuse and derivative use of Label Generation Policy. .

The team further suggests that the ASTT , which defines the acceptable string input code points for a TLD label at the root, needs to be more conservative compared to other levels.

However, when this ASTT is defined, it should be combined with a PDP (Policy development Procedure) for updating and handling community comments.

1. Though defining the structure in necessary for effective re-use of this information, it was considered out of scope of the current work on issues.

One issue here regading wether variants should be defined at the character positioning level (Initial, Medial, Final, Isolated) or at character level Positioning level gives more choices to the users, without making them any more confusable. The character level is a subset of positioning level. If the former is made available, the latter can be inherited as well (but not vice versa). This will have implications on the design of the structure for the ASTT.

1. **Registration Process**

Introduction of variants causes more complexity in the registration process, as there are many more strings which are generated as variant labels and not all of these may be eventually delegated. Should there be a limit on the size of the Variant Label Set, else otherwise the list may be quite long? If yes, how should the short-list be determined?

A registrant would normally request a Fundamental Label. However, now a Variant Label Set will be generated as per the Label Generation Policy (this is fixed by the registry policy and not determined by the registrant). All these labels, except the Blocked Variant Label Subset (which is also determined by registry policy), will be available for the registrant. As per the decision of the registrant and the policies of the registry, the available labels will be divided into Allocated Variant Label Subset and Reserved Variant Label Subset, former administratively associated with the registrant and latter not available to anybody else but not administratively associated to the registrant. All the Allocated Variant Labels may eventually be activated through delegation or other mechanisms (e.g. DNAME or any other methods available). If only a subset is activated then this also need to be defined.

Also over time, the registrant may request for changing the status of one or more of the Variant Labels, e.g. from allocated to reserved, vice versa, etc.

It needs to be clearly defined what are the multiple states a Variant Label may take and what are the processes to transition from a state to another state, including administrative, technical and billing requirements. For example, it is recommended that the allocation and expiration of Variant Label Set be synchronized, and the transition process for individual labels be possible within these time bounds. Another associated question is whether history of such transitions be maintained?

Another important consideration is about how this process will be different if the registrant request change in status of a fundamental label? Will this be possible (should be, if the variants are symmetrical in Arabic script)? This may be more complex to deal with as an alternate fundamental label will need to be defined and other variant labels will need to be re-associated (e.g. see the Domain Name Registration Data (WHOIS) section).

If the domain name expires or is transferred to somebody else, how will the Variant Label Set change its status? In the case of transfer, will the transferee get the option to choose of various subsets differently from what exist at this time (fundamental, allocated, reserved, etc.)? This is applicable to TLDs, in cases of re-delegation.

Additionally, these need to be articulated clearly for registrants to understand the implications and to make informed decisions.

The current registration tools need to be extended to incorporate the entire process automatically, with no (or minimal) manual interaction.

Finally, it is possible to change the Language Generation Policy. As it is changed, it may modify the Variant Label Set. How will such changes and their impact be communicated to the stakeholders, e.g. the registrant (with now additional variant labels which may be available as Unicode assigns an additional letter in its repertoire).

1. **Domain Name Registration Data (WHOIS)**

As the domain names become available in Arabic script, there is clear need for the Domain Name Registration Data to be in the same script. The current WHOIS Protocol (RFC 3912) does not support multilingual data. There are already some efforts in the community to come up with a more comprehensive protocol which also supports this aspect and this should be pursued and finalized.

In addition to the Domain Name Registration Data Protocol[[4]](#footnote-6), it should also be clearly defined what Domain Name Registration Data is needed from the registrant and how much of that will be made available to the public.

It is also crucial to define what Domain Name Data Services are needed to make this non-ASCII data available publically.

As the protocol, data and service needs around the IDNs are being finalized, the support for variants must also be considered. There is a paradigm shift here, where though currently there is a one to one lookup for WHOIS against the domain name, now the look up has to deal with the following scenario, where Label Xi represents a variant of Label Xk.

|  |  |  |
| --- | --- | --- |
| Label 31  **…**  Label 3l | Label 21  **. …**  Label 2m | Label 11  **. …**  Label 1n |

There are multiple ways variants may be queried, and the possible domain names may have different statuses (e.g. reserved, allocated, delegated, activated, etc.).

It needs to be determined how a request for domanin name Label3i.Label2j.Label1k may respond.

The Arabic script case study team suggests the following:

* All variants should have the same ownership
* It would be necessary to have the variant information available as part of the Domain Name Registration Data Services
* A query on an allocated or reserved label should return the fundamental label; though it is not clear at this time if the status of the queried label should also be returned along with the fundamental label.
* A query on the fundamental label would return the Domain Name Registration Data relevant to the query. Details of what is returned need to be defined (as discussed above).
* There is need for an additional query/service which returns the Label Variant Set against a requested domain name. Again it is not clear, if such a service should also return the status of each label in the set.
* Would the response against a blocked variant label different from responses to labels with other status (reserved, allocated, etc.)?
* Will the creation and expiration of the Variant Label Set be inherited from the fundamental label, as suggested? If yes, then if a variant label is either added or changes its state, how will this information be part of the Domain Name Registration Data? Would history be maintained and communicated for such changes?

1. **Blocked Variant Label Set (for Reserved Names)**

The registry may need to block reserved names and their variants (e.g. geographic names). Here are some relevant questions.

1. If someone gets an exception to register a name from the reserved list, should s/he also get its variants? If yes, does this happen automatically or upon request?
2. Should the registry add all variants or some of them to the reserve list? How is it determined?
3. **Fees**

As variant labels introduce a much more complex mechanism, with a process to change the status of various labels, the implication on pricing should also be developed. Relevant questions include the following:

* 1. Does the fee differ with the size of the Label Variant Set requested to be Allocated?
  2. If it is possible to activate differently, are there different fees for these processes, e.g. Delegation vs. DNAME, etc.
  3. Is there a fee for changing the status of a variant from one status to another, e.g. from reserved to delegated, etc.?

1. **DNSSEC**

The registry may need to think more about key management specially when they adopt variants. Delegation may require different keys, while DNAME like techniques may require a single key. A mixture of such processes may require a complex key management mechanism. This may also cause different expiry dates for different labels in the same Variant Label Set and may be needed to be looked carefully when transferring the domain or changing the status of a variant label.

1. **Dispute Resolution**
   * 1. **Filing a Complaint/Case**
        1. Are bundled domain names treated as one also with regards to dispute resolution?
        2. Could a case exist where a complainer claims right over a member (a label) from within a bundle? Will the dispute differ if the disputed domain name is active, blocked or reserved? Will the whole bundle be somehow included in this case?
        3. Could a case exist where a complainer claims right over a whole bundle? Will all bundle members be consolidated as one case?
        4. Could a case exist where a complainer claims right over a bundle containing members/labels in different languages? Within the same script? Across different scripts?
        5. How to prove ‘Bad Faith’ in case of inactive domain names (i.e. blocked or reserved within a bundle)?
        6. May registrants be forced into disputes as a result of automated bundling?
        7. Could a case be filed against a domain name that is still under request or does it have to be delegated?
     2. **Payment**
        1. How would the following different cases affect the fee scheme, i.e. filing a case regarding a whole bundle, one member/label that belongs to a bundle, more than one member/label that belongs to a bundle?
        2. Could the fee differ according to the status of the disputed domain name whether it is active, blocked or reserved?
     3. **Decision**
        1. Could a case exist where a decision (such as cancellation or transfer of ownership) is applied to only one member/label of a bundle? If yes, what are the implications? In case the rest of the members were not active does this give the right to the registrant to activate any one or more members of the bundle? Or is the winner allowed to assume ownership of the whole bundle?
        2. With respect to disputes, are blocked or reserved labels considered domain name in use?
        3. Could a case exist where a domain name is added to an existing bundle, either because it was overlooked in the pre-defined sets or not opted by the original registrant?
        4. Is it possible to have a single dispute spanning two different bundles?
        5. Can a UDRP decision have implications on a language table (i.e. can a UDRP decision cause change changes to an existing variant table?)
        6. Do the above cases assume extending the UDRP scope beyond trademarks?
        7. Could a case exist where two trademarks in two different languages, within the same script (Arabic in our case), be confusingly similar; i.e. should be treated as a bundle according to our definition of a bundle whereas from a trademark perspective they have two different eligible owners?
        8. Should a change in a domain name status, due to its being under dispute, be inherited by the rest of the bundle members
     4. **Technical Implementation of a Dispute Resolution Decision:**
        1. What are the implications of changes done on a variant set as a result of implementing a dispute resolution decision?
        2. What are the implications of a dispute resolution decision on Variant Tables in terms of implementing changes to variant sets defined. Can a decision on a bundle have impact on other bundles due to implied changes in the variant table?
2. **End-user Requirements**

Labels and their variants will need to be configured and used by administrators and users respectively. Appropriate tools and applications need to be developed to assist the process.

* 1. **Keyboard (soft) issues**
     1. Lack of standard keyboard for many languages using Arabic script (e.g. for Kurdish, Urdu, Pashto, Sindhi and many other languages). This is also true when a user travels across different countries, trying to access, for example, Farsi domain name from an Arabic language country (with an Arabic KB available locally). This causes variation in typing same labels, and thus requires activation of labels for effective use for Arabic script.
     2. Digits may also present an issue. But not relevant to TLDs.
  2. **Font issues**
     1. Two very different writing styles are in use by Arabic script community. Arabic language uses Naskh writing style and Persian, Urdu, Pashto, and many other languages in South Asia use Nastalique writing style. There are ;many other font styles in use, including Thuluth, Riqa, Qufi and others, for stylistic variation. Fonts may have implications on variant label sets.
     2. Font support may not be consistent across variant label set, as a font may not support all characters variants, causing strings to break during display or boxes or other unexpected characters to show up. This may cause user confusion by not making different variants visible, and by making labels which are not variants to appear as same (e.g. two different letters mapped onto a box making them indistinguishable).
  3. **Concurrent Display of A-label and U-label for Administrative Purposes**
     1. Tool need to develop to manage variant labels. For example, currently there are no EPP extensions to handle variants during the registration process.
     2. Similarly the administration tools only deal with ASCII, meaning that server administration will only be possible using A-Labels. Tools need to be developed to view these A-Labels along with the U-Labels to avoid management errors and to better identify the variants.
     3. The two cases above only present a examples of many other tools which need to be developed for different functions and which use domain names and labels. A more exhaustive list needs to be worked out and eventually addressed. Though these are not relevant to TLDs, they are certainly relevant to their eventual use in the domain name system.
  4. **Bidirectionality Issues**
     1. Arabic script domain name may have characters of different directionality (Arabic letters, Arabic digits, Arabic Indic digits, ASCII LDH, etc.) mixed within a label and across labels. This causes inconsistent display across various applications and may confuse users. Such cases need to be investigated and displayed in a consistent manner in applications.
  5. **Other issues**
     1. Operating System support may not be consistent across variants of the same label, as an OS may not support all variant characters in a language table, causing strings to break during display or boxes or other unexpected characters to show up. This may cause user confusion by not making different variants visible, and by making labels which are not variants to appear as same (e.g. two different letters mapped onto a box making them indistinguishable).

1. **List of relevant stakeholders**
2. **Concluding Remarks**
3. **References**

M. Davis and M. Durst (1999). Unicode Normalization Forms. Accessed from <http://unicode.org/reports/tr15/tr15-18.html>**. Appendices**

**Appendix A. Identical Character Variants**

**Appendix A.1. Same Shape in at least one Position**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unicode** | **Initial Form** | **Medial Form** | **Final Form** | **Isolated Form** |
| Kaf Group | | | | |
| **U+06A9 (ک)** | کا | لکل | ٹک | ک |
| **U+0643 (ك)** | كا | لكل ل | ٹك | ك |
| Hay Group | | | | |
| **U+0647 (ه)** | هم | مها | له | ه |
| **U+06BE (ھ)** | ھم | مھا | للھ | ھ |
| **U+06C1 (ہ)** | ہم | مہا | کہ | ہ |
| **U+06D5 (ە)** | - | - | نە | ە |
| Yay Group | | | | |
| **U+064A (ي)** | يع | ميل | یلي | ي |
| **U+06CC (ی)** | یع | میل | یلی | ی |
| **U+0649 (ى)** | - | - | بى | ى |
| Fay Group | | | | |
| **U+06A7 (ڧ)** | ڧر | کڧر | یڧ | ڧ |
| **U+0641 (ف)** | فر | کفر | یف | ف |
| Veh Group | | | | |
| **U+06A4 (ڤ)** | ڤر | کڤر | یڤ | ڤ |
| **U+06A8 (ڨ)** | ڨر | کڨر | لڨ | ڨ |
| Tay Marbuta Group | | | | |
| **U+0629 (ة)** | - | - | بة | ة |
| **U+06C3 (ۃ)** | - | - | بۃ | ۃ |
| Hay with Hamza Group | | | | |
| **U+06C0 (ۀ)** | - | - | بۀ | ۀ |
| **U+06C2 (ۂ)** | - | - | بۂ | ۂ |
| Ttey/Rnoon Group | | | | |
| **U+06BB (ڻ)** | ڻا | بڻن | بڻ | ڻ |
| **U+0679 (ٹ)** | ٹا | مٹن | بٹ | ٹ |
| Noon Group | | | | |
| **U+0646 (ن)** | نا | بنن | بن | ن |
| **U+06BA (ں)** | ںا | مںن | بں | ں |
| Theh Group | | | | |
| **U+06BD (ڽ)** | ڽا | مڽا | ڽ | ڽ |
| **U+067E (پ)** | پا | بپا | پ | پ |

**Appendix A.2. Same Shape in Composed and Decomposed forms using Combining Marks**

| **Combining Mark** | **Composed Form** | **Decomposed Form** | **Unicode Normalized Form** |
| --- | --- | --- | --- |
| **ٓ**  U+0653 | آ  U+0622 | ٓ ا  U+0627 U+0653 | Defined |
| ٔ  U+0654 | أ  U+0623 | ٔ◌ ا  U+0627 U+0654 | Defined |
| ؤ  U+0624 | ٔ و  U+0648 U+0654 | Defined |
| ئ  U+0626 | ٔ ي  U+064A U+0654 | Defined |
| ىٔ  U+0649 U+0654 | Not Defined |
| یٔ  U+06CC U+0654 | Not Defined |
| ۀ  U+06C0 | ۀ  U+06D5 U+0654 | Defined |
| هٔ  U+0647 U+0654 | Not Defined |
| ۂ  U+06C2 | ۂ  U+06C1 U+0654 | Defined |
| هٔ  U+0647 U+0654 | Not Defined |
| ۓ  U+06D3 | ۓ  U+06D2 U+0654 | Defined |
| ځ  U+0681 | حٔ  U+062D U+0654 | Not Defined |
| U+076C | رٔ  U+0631 U+0654 | Not Defined |
| ٕ  U+0655 | إ  U+0625 | ا◌ٕ  U+0627 U+0655 | Defined |
| ُ  U+064F | ۇ  U+06C7 | ُ و  U+0648 U+064F | Not Defined |
| U+0648 U+0619 | Not Defined |
| ٰ  U+0670 | ۈ  U+06C8 | ٰ و  U+0648 U+0670 | Not Defined |
| ۬  U+06EC | ۏ  U+06CF | ۬ و  U+0648 U+06EC | Not Defined |
| غ  U+063A | ۬ ع  U+0639 U+06EC | Not Defined |
| ض  U+0636 | ۬ ص  U+0635 U+06EC | Not Defined |
| خ  U+062E | ۬ ح  U+062D U+06EC | Not Defined |
| ڿ  U+06BF | ۬ چ  U+0686 U+06EC | Not Defined |
| ذ  U+0630 | ۬ د  U+062F U+06EC | Not Defined |
| ز  U+0632 | ۬ ر  U+0631 U+06EC | Not Defined |
| ڶ  U+06B6 | ۬ ل  U+0644 U+06EC | Not Defined |
| ڧ  U+06A7 | ٯ۬  U+066F U+06EC | Not Defined |
| ف  U+0641 | ۬ ڡ  U+06A1 U+06EC | Not Defined |
| ن  U+0646 | ۬ ں  U+06BA U+06EC | Not Defined |
| ڬ  U+06AC | ۬ ك  U+0643 U+06EC | Not Defined |
| U+0762 | ۬ ک  U+06A9 U+06EC | Not Defined |
| U+0765 | ۬ م  U+0645 U+06EC | Not Defined |
| U+0615 | U+0772 | ح  U+062D U+0615 | Not Defined |
| ٹ  U+0679 | ٮ  U+066E U+0615 | Not Defined |
| ڑ  U+0691 | ر  U+0631 U+0615 | Not Defined |
| ڈ  U+0688 | د  U+062F U+0615 | Not Defined |
| U+0771 | ڗ  U+0697 U+0615 | Not Defined |
| U+0768 | ن  U+0646 U+0615 | Not Defined |
| ڋ  U+068B | ڊ  U+068A U+0615 | Not Defined |
| ڻ  U+06BB | ں  U+06BA U+0615 | Not Defined |
| U+065B | U+063D | ی  U+06CC U+065B | Not Defined |
| ۉ  U+06C9 | و  U+0648 U+065B | Not Defined |
| U+077E | س  U+0633 U+065B | Not Defined |
| U+06EE | د  U+062F U+065B | Not Defined |
| U+06EF | ر  U+0631 U+065B | Not Defined |
| U+06FF | ھ  U+06BE U+065B | Not Defined |
| U+0647 U+065B | Not Defined |
| ۛ  U+06DB | U+063F | ۛ ی  U+06CC U+06DB | Not Defined |
| ۛ ى  U+0649 U+06DB | Not Defined |
| ش  U+0634 | ۛ س  U+0633 U+06DB | Not Defined |
| ڜ  U+069C | ۛ ڛ  U+069B U+06DB | Not Defined |
| ث  U+062B | ٮۛ  U+066E U+06DB | Not Defined |
| څ  U+0685 | ۛ ح  U+062D U+06DB | Not Defined |
| ژ  U+0698 | ۛ ر  U+0631 U+06DB | Not Defined |
| ڎ  U+068E | ۛ د  U+062F U+06DB | Not Defined |
| ڠ  U+06A0 | ۛ ع  U+0639 U+06DB | Not Defined |
| ڤ  U+06A4 | ۛ ڡ  U+06A1 U+06DB | Not Defined |
| ڨ  U+06A8 | ٯۛ  U+066F U+06DB | Not Defined |
| ڭ  U+06AD | ۛ ك  U+0643 U+06DB | Not Defined |
| ڴ  U+06B4 | ۛ گ  U+06AF U+06DB | Not Defined |
| ڷ  U+06B7 | ۛ ل  U+0644 U+06DB | Not Defined |
| ڽ  U+06BD | ۛ ں  U+06BA U+06DB | Not Defined |
| U+0763 | ۛ ک  U+06A9 U+06DB | Not Defined |
| U+065C | ب  U+0628 | ٮ  U+066E U+065C | Not Defined |
| ڊ  U+068A | د  U+062F U+065C | Not Defined |
| ڋ  U+068B | ڈ  U+0688 U+065C | Not Defined |
| ڔ  U+0694 | ر  U+0631 U+065C | Not Defined |
| ڣ  U+06A3 | ف  U+0641 U+065C | Not Defined |
| ڹ  U+06B9 | ن  U+0646 U+065C | Not Defined |
| ۼ  U+06FC | غ  U+063A U+065C | Not Defined |
| ۻ  U+06FB | ض  U+0636 U+065C | Not Defined |
| U+0751 | ث  U+062B U+065C | Not Defined |
| U+0766 | م  U+0645 U+065C | Not Defined |
| U+065A | ڵ  U+06B5 | ل  U+0644 U+065A | Not Defined |
| ۆ  U+06C6 | و  U+0648 U+065A | Not Defined |
| ێ  U+06CE | ی  U+06CC U+065A | Not Defined |
| ى  U+0649 U+065A | Not Defined |
| U+0756 | ٮ  U+066E U+065A | Not Defined |
| U+0769 | ن  U+0646 U+065A | Not Defined |

**Appendix A.2.2. Same Shape in Composed and Decomposed forms using Two Combining Marks**

|  |  |
| --- | --- |
| Composed Form | Decomposed Form |
| ښ  U+069A | س  U+0633 U+065C U+06EC |
| ڣ  U+06A3 | ڡ  U+06A1 U+065C U+06EC |
| ۺ  U+06FA | س ۛ  U+0633 U+06DB U+065C |
| ۻ  U+06FB | ص  U+0635 U+065C U+06EC |
| ۼ  U+06FC | ع  U+0639 U+065C U+06EC |
| ڹ  U+06B9 | ں  U+06BA U+065C U+06EC |

**Appendix B. Confusable Similar Letters in Arabic Script**

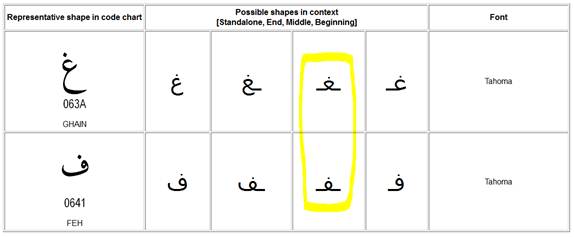
**Appendix B.1. Similar Shape in at least one Position**

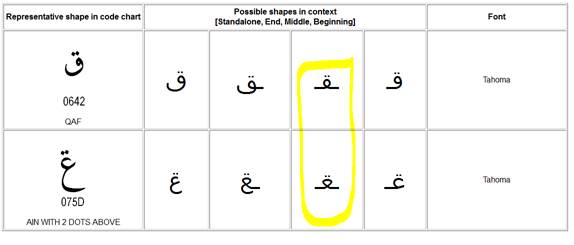
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unicode** | **Initial Form** | **Medial Form** | **Final Form** | **Isolated Form** |
| Kaf Group | | | | |
| U+06A9 (ک) | کا | لکل | ٹک | ک |
| U+06AA (ڪ) | ڪ | لڪل | ٹڪ | ڪ |
| U+0643 (ك) | كا | لكل ل | ٹك | ك |
| Yay Group | | | | |
| U+064A (ي) | يع | ميل | یلي | ي |
| U+06CC (ی) | یع | میل | یلی | ی |
| U+0649 (ى) | - | - | بى | ى |
| U+06CD (ۍ) | - | - | لۍ | ۍ |
| Hamza Above Group | | | | |
| U+0623 (أ) | - | - | بأ | أ |
| U+0672 (ٲ) | - | - | بٲ | ٲ |
| Hamza Below Group | | | | |
| U+0625 (إ) | - | - | بإ | إ |
| U+0673 (ٳ) | - | - | بٳ | ٳ |

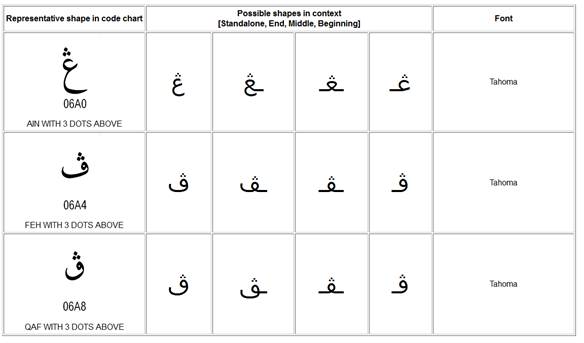
**Appendix B.2. Confusable Similar Shape with Difference in Dot Orientation**

|  |  |
| --- | --- |
| **Unicode** | **Characters** |
| 1. U+062A 2. U+067A | 1. ت 2. ٺ |
| 1. U+062B 2. U+067D | 1. ث 2. ٽ |
| 1. U+063C 2. U+0764 | 2. ݤ |
| 1. U+064A 2. U+06D0 | 1. ي 2. ې |
| 1. U+067E 2. U+0752 | 1. پ 2. ݒ |
| 1. U+0683 2. U+0684 | 1. ڃ 2. ڄ |
| 1. U+0686 2. U+0758 | 1. چ 2. ݘ |
| 1. U+068E 2. U+068F | 1. ڎ 2. ڏ |
| 1. U+06A0 2. U+075F | 1. ڠ 2. ݞ |
| 1. U+06B2 2. U+06B3 | 1. ڲ 2. ڳ |
| 1. U+075D 2. U+075E | 1. ݝ 2. ݟ |
| 1. U+0697 2. U+076B | 1. ڗ 2. ݫ |

**Appendix B.2. Possibly Confusable Similar Shape with in Small Font Sizes**







1. Definitions accessible at <http://www.icann.org/en/topics/new-gtlds/idn-implementation-working-team-report-final-03dec09-en.pdf> [↑](#footnote-ref-1)
2. See ICANN Board of Directors. (2010) Adopted Board Resolutions. Trondheim, Norway. Retrieved November 30, 2010, from <http://www.icann.org/en/minutes/resolutions-25sep10-en.htm#2.5> [↑](#footnote-ref-2)
3. Unicode standard also include Arabic Presentation forms in the ranges U+FB50 – U+FDFF and U+FE70 – U+FEFF, which are not recommended for use and are not PVALID. [↑](#footnote-ref-4)
4. This section is using the terminology being recommended by SSAC Report on Domain Name (WHOIS) Terminology and Structure (not published at the time of this report) [↑](#footnote-ref-6)